

Ville-Pekka Tuomikoski

Metsäkoneessa käytettävien SQL-tietokantojen päivitys

Insinööri (AMK)

Tieto- ja viestintätekniikka

Kevät 2018



KAJAANIN
AMMATTIKORKEAKOULU
UNIVERSITY OF APPLIED SCIENCES

Tiivistelmä

Tekijä(t): Tuomikoski Ville-Pekka

Työn nimi: Metsäkoneessa käytettävien SQL-tietokantojen päivitys

Tutkintonimike: Insinööri (AMK), tieto- ja viestintätekniikka

Asiasanat: SQL, relaatiotietokannat

Työn tilaajana oli Devecto Oy:n Kajaanin toimisto ja sen asiakas Ponsse Oyj, tarkemmin Ponssen Kajaanin tuotekehitys. Devecto Oy, entiseltä nimeltään Vision Development Oy, on Jyväskylästä lähtöisin oleva yritys, jonka toimintaan kuuluu muun muassa sulautettu ja mallipohjainen ohjelmistosuunnittelu ja ohjelmistotestaus. Ponsse Oyj on vieremäläinen metsäkoneyritys, joka on yksi metsäkonealan markkinajohtajista ja jonka tuotekehitys tapahtuu Kajaanissa.

Työn tavoitteena oli löytää jokin keino päivittää metsäkoneissa käytettäviä SQL-relaatiotietokantoja metsäkoneen mittalaitteen ohjelmiston päivityksen yhteydessä niin, ettei tietokannoissa oleva tieto häviä tai muutu päivityksessä. Työssä oli tarkoituksena tutkia olemassaolevia työkaluja ja tutkia, pystyykö niitä hyödyntämään tietokantojen päivityksessä.

Työssä tutkittiin pääasiassa Redgate Softwaren valmistamia työkaluja ja hyödynnettiin Ponsselta jo valmiina olevaa koodia asiaan liittyen. Työkaluilla pystyi tekemään skriptejä, joilla tietokannat pystyi päivittämään. Työkaluja tutkittiin Redgaten dokumentteja lukemalla ja testattiin sekä omalla tietokoneella että Ponssen simulaattoreissa kokeilemalla, mitä muutoksia työkalut osaavat tietokantoihin päivittää. Havaittiin, että työkalut selvittävät suurimman osan erilaisista tietokantamuutoksista ongelmitta.

Työn tuloksena saatiin selvyys, mikä työkalu soveltuu työhön parhaiten. Lisäksi pystyttiin demonstroimaan tekniikkaa, jolla tietokannat voidaan päivittää, mutta aivan valmista ratkaisua ei saatu tehtyä. Työ vaatii jatkokehitystä, jotta työkalut voidaan ottaa mukaan Ponssen kehitysrutiiniin. Lisäksi täytyy ratkaista ongelma, joka estää tiettyjen tietokantamuutosten tekemisen.

Abstract

Author(s): Tuomikoski Ville-Pekka

Title of the Publication: Updating SQL Databases Used in Forestry Machines

Degree Title: Bachelor of Engineering, Information Technology Engineering

Keywords: SQL, relational database

This thesis was commissioned by Devecto Oy's office in Kajaani, Finland and their customer Ponsse Oyj's research and development unit in Kajaani. Devecto Oy, formerly known as Vision Development Oy, was founded in Jyväskylä, Finland, 2014 and it develops embedded and model based software solutions and software testing. Ponsse Oyj, originating from Vieremä, Finland, is one of world's leading forest machine manufacturers and its research and development unit is located in Kajaani.

The purpose of this thesis was to find a solution to automatically update SQL databases in Ponsse's forestry machines while performing software update in the machines. Because the information in the databases is valuable to Ponsse's customers, one requirement for the thesis was to make sure that customers' data won't be damaged while committing the update. The work consisted mainly of researching existing tools made for the job and testing them to see if they work for the purposes of this thesis.

The result of the thesis was a demonstration of how to use the tools to update the databases, but a full solution was not yet reached. ReadyRoll was found to be the best tool for updating databases. Further research and development is needed to bring the tools into use at Ponsse's regular development routine regarding databases.

Alkusanat

Haluan kiittää Devecto Oy:n Petri Ingalsuota ja Ponsse Oyj:n Tomi Piiraista tämän insinööriyön mahdollistamisesta ja työhön liittyvästä avusta. Lisäksi haluan kiittää erityisesti Ponsse Oyj:n Sami Ahosta kaikesta työn aikana saadusta avusta. Kiitos myös kaikille muille, jotka auttoivat ja kannustivat minua työssä.

Sisällys

1	Johdanto	1
2	SQL ja tietokannat.....	2
2.1	Microsoft SQL Server	4
2.2	SQL Server Management Studio (SSMS)	4
2.3	T-SQL	4
3	Käytetyt työkalut.....	5
3.1	Microsoft Visual Studio 2015	5
3.2	Redgate SQL Compare.....	5
3.3	Redgate SQL Data Compare	6
3.4	Redgate ReadyRoll	7
3.5	InstallShield.....	7
4	Metsäkoneen mittalaite Ponsse Opti sekä Opti4G.....	8
5	Työn toteutus	9
5.1	Vaatimukset työlle	9
5.2	Työn suunnittelu.....	9
5.3	Työn toteutus	9
5.4	Työn tulokset.....	16
	Lähteet.....	18
	Liitteet	

Symboliluettelo

DLL	Dynamic Link Library on jaettu kirjasto, joka sisältää koodia, jota muut ohjelmat voivat käyttää.
SQL	Structured Query Language on ohjelmointikieli, joka on suunniteltu tietojen keräämiseen tietokannoista ja tiedon käsittelyyn tietokannoissa.
StanForD	Standard for Forest machine Data and Communication on metsäkoneiden tiedonsiirtoon kehitetty standardi.
T-SQL	Transact-SQL on Sybasen ja Microsoftin kehittämä laajennus SQL:ään.

1 Johdanto

Hakkuukoneille on kehitetty tiedonsiirtostandardi StanForD, joka on lyhenne sanoista "Standard for Forest machine Data and Communication". Se on siis standardi, joka määrittelee tiedonsiirron tietokoneiden ja metsäkoneiden välillä. Tähän standardiin tulee muutoksia säännöllisin väliajoin, ja ne saattavat aiheuttaa metsäkoneyhtiöille tarpeen päivittää metsäkoneissaan olevia tietokantoja uuteen standardiin sopiviksi.

Insinööriyön tavoitteena oli tutkia, miten Ponssen metsäkoneiden mittalaitteilla olevat paikalliset SQL-tietokannat voidaan päivittää. Tietokantojen päivityksen tuli tapahtua metsäkoneen mittalaitteen ohjelmistopäivityksen yhteydessä.

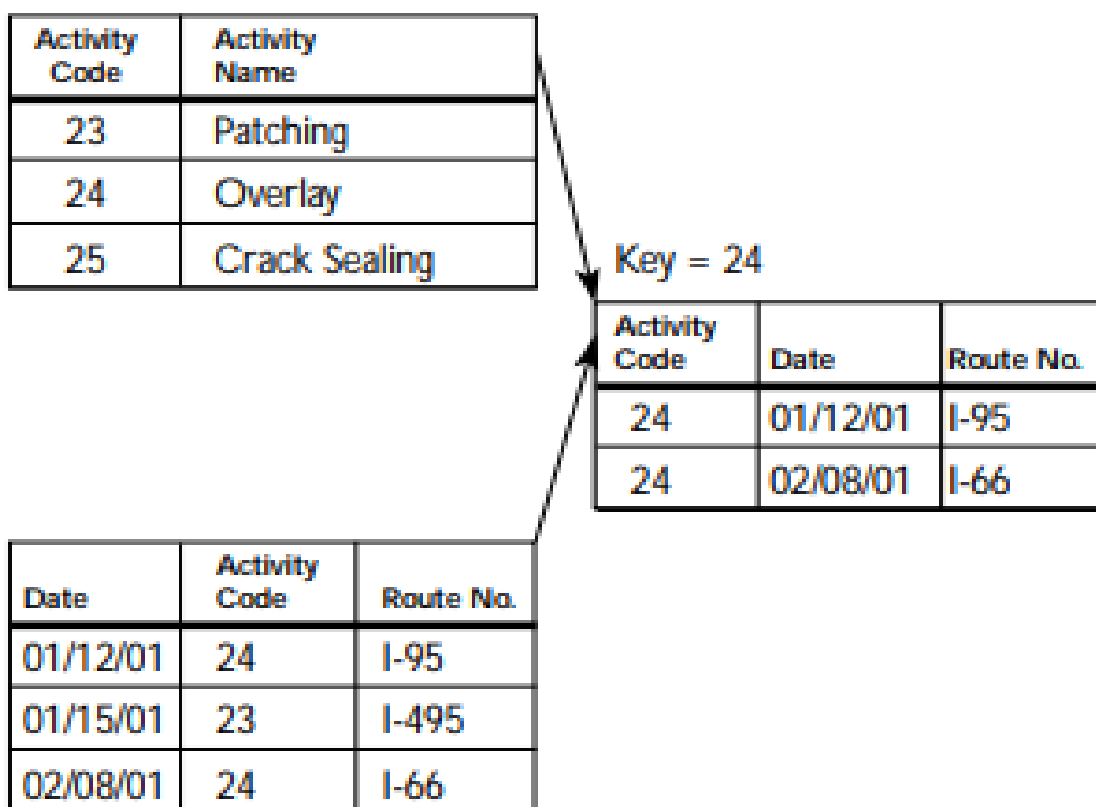
Työn toimeksiantajana toimi Devecto Oy, ja työ tehtiin Devecton asiakkaalle Ponsse Oyj:n Kajaanin tuotekehitykselle sen tiloissa. Devecto Oy, entiseltä nimeltään Vision Development Oy, on Jyväskylästä lähtöisin oleva yritys, jonka toimintaan kuuluu muun muassa sulautettu ja mallipohjainen ohjelmistosuunnittelu ja ohjelmistotestaus. Ponsse on suomalainen metsäkoneyhtiö, joka perustettiin vuonna 1970 Einari Vidgrénin toimesta. Ponssen metsäkoneet on kehitetty pitäen mielessä tavaralajimenetelmä, jossa puiden rungot katkotaan jo metsässä tarkoituksenmukaisiin mittoihin. Tietojärjestelmät ovat metsäkoneissa tärkeässä osassa, sillä ne tallentavat ja ilmoittavat asiakkaalle, minkä verran ja millaista puuta metsässä on hakattu. [1.]

Yhtenä vaatimuksena työlle oli, että tietokantojen päivityksen tulee peruuntua, jos ohjelmaversion päivitys epäonnistuu, koska tietty tietokantaversio toimii vain tietyn ohjelmistoversion kanssa. Ohjelmaversion päivitys tapahtuu InstallShieldillä tehdyllä asennusohjelmalla. Toisena vaatimuksena oli myös tietokantaan tallennettujen tietojen säilyminen muuttumattomana päivityksen yli, sillä tietokannassa on asiakkaille rahanarvoista tietoa ja sitä ei saa kadottaa tai muuttaa.

2 SQL ja tietokannat

SQL eli Structured Query Language on IBM:n (International Business Machines Corporation) 1970-luvulla kehittämä ohjelmointikieli, jota käytetään relaatiotietokantojen hallintaan. SQL-kielillä voidaan tehdä hakuja, muutoksia ja lisäyksiä tietokantoihin. SQL-kieli muistuttaa englannin kieltä: esimerkki komennosta voisi olla "SELECT * FROM CUSTOMERS", jolla voidaan valita kaikki sisältö taulukosta, jonka nimi on "CUSTOMERS". [2.]

Tietokanta on tietyn logiikan mukaan järjestelty kokoelma tietoa. Yleisimmin käytetty tietokantamalli on relaatiotietokantamalli. Muita tietokantamalleja ovat oliotietokantamalli ja olio-relaatiotietokantamalli, joka on edellisten mallien yhdistelmä. Relaatiotietokannassa tieto organisoidaan taulukoittain ja taulujen välille voidaan luoda yhteyksiä. Näin samaa tietoa voi käyttää monessa eri taulukossa, ja jos tietoa halutaan muokata, se muuttuu myös muissa taulukoissa. Kuvassa 1 on relaatiomallin mukainen taulu, joka yhdistää saman avaimen alla olevat tiedot kahdesta taulusta kolmanteen yksittäiseen tauluun. Yhdistetty taulu on siistimpi, kuin jos kaikki tiedot olisivat samassa taulussa. Tämä luonnollisesti auttaa huomattavasti tietokannan ylläpidossa. [3.]



Kuva 1. Esimerkki relaatiotietokantamallin mukaisesta tietokantakaaviosta [4].

Kun työssä puhutaan tietokantojen päivittämisestä, ei tarkoiteta tietokannassa olevan tiedon päivittämistä, vaan tietokantarakenteen päivittämistä. Mahdollisia päivitettäviä asioita ovat muun muuassa taulujen (table), proseduurien (stored procedure), rajoitteiden (constraint) ja indeksien (index) lisäys, muokkaus ja poisto.

Proseduurit ovat funktioita, joita tietokantaan voi halutessaan tehdä. Proseduuri voi ottaa syötteenään parametreja, ja se voi palauttaa yhden tai useampia arvoja, kun sitä kutsutaan. Proseduurien avulla voidaan vähentää palvelimelle tehtävien kutsujen määrää, sillä proseduuri voi sisältää useita rivejä koodia. Jos proseduuria ei olisi olemassa, nuo koodirivit pitäisi lähettää palvelimelle joka kerta, kun ne halutaan suorittaa. Sen lisäksi, että kutsut palvelimelle vähentyvät, muistissa olevan proseduurin suoritus on nopeampaa kuin ikään kuin yllätyksenä saapuvan koodin suoritus. Proseduureja voi muokata ilman, että se vaikuttaa tietokannassa oleviin tietoihin, joten niiden päivityksen ei pitäisi olla ongelma tässä työssä. [5.]

Rajoitteet ovat kentille lisättäviä valinnaisia rajoittavia ehtoja. Ne rajaavat, millaista tietoa kenttään voi lisätä. Jos kenttään tallennetaan vaikkapa numeroita, voidaan rajata, että siihen saa tallentaa numeroita vain väliltä 0–100 tai että kenttään ei saa tallentaa tyhjää arvoa. Rajoitteiden päivityksessä on haasteena ainakin se, että jos tietokantaan on tallennettu tietoa, joka ei ole uuden rajoitteen sisällä, rajoitteen lisäys ei ole aina mahdollista. Päivityksessä tulisikin olla aivan varma, ettei rajoitteiden ulkopuolella olevaa tietoa ole, tai sitten tietoa on pakko muokata ennen päivityksen suorittamista. [6.]

Indeksejä voi käyttää tietokannan hakujen nopeuttamiseen. Niitä voi verrata kirjan sisällysluetteloon, ne tavallaan kertovat tietokannalle, mistä indeksin osoittama tieto löytyy. Indeksit nopeuttavat tiedon hakua "SELECT"- ja "WHERE"-komennoilla, mutta hidastavat tiedon lisäämistä ja päivitystä eli "INSERT"- ja "UPDATE"-komentoja. Indeksejä voidaan luoda ja poistaa, eikä se vaikuta tietokannassa olevaan tietoon, eli indeksien päivityksen tulisi vaivatta onnistua helposti ja turvallisesti. [7.]

2.1 Microsoft SQL Server

Microsoftin kehittämä Microsoft SQL Server on tietokantojen hallintajärjestelmä. Sen avulla tietokantapalvelimia ja tietokantoja voidaan luoda käyttöä varten. Monella sovelluksella voi olla yhteys samaan tietokantaan, ja ne voivat muokata sitä samanaikaisesti. Tietokantaan voidaan luoda monia eri käyttäjiä, joille voidaan antaa oikeudet eri asioihin. Jos esimerkiksi jollakin sovelluksella tarvitsee vain lukea tietoja, sille voidaan tehdä oma käyttäjä, jolla on vain lukuoikeudet. Työssä täytyi ottaa huomioon, että Ponssella oli käytössä Microsoft SQL Server 2014 Express 32-bit, jossa tietokannan maksimikoko on 10 GB. Se on myös muilta ominaisuuksiltaan rajoitetumpi kuin Enterprise-, Standard- ja Developer-versiot. [8.]

2.2 SQL Server Management Studio (SSMS)

Microsoftin kehittämä SQL Server Management Studio eli SSMS on Microsoftin SQL-palvelimien ja SQL-tietokantojen ylläpitoon tarkoitettu työkalu [9]. Sillä voi muokata ja tarkastella SQL-tietokantoja. Tässä työssä SSMS-työkalua käytettiin välillä apuna tietokantamuutosten tekemisessä ja tietokannassa olevan tiedon tarkasteluun ja muokkaamiseen.

2.3 T-SQL

T-SQL eli Transact-SQL on Microsoftin ja Sybasen kehittämä laajennus SQL:ään. T-SQL tuo mukanaan SQL-lauseiden suoritusten valvonnan, poikkeuksien ja virheiden käsittelyn ja muuttujien käyttämisen. Sovellukset, jotka kommunikoivat Microsoftin SQL Serverin kanssa, käyttävät siihen T-SQL:ää. Työssä tutkitut Redgaten työkalut, joilla pystyi generoimaan tietokannan päivitysskriptejä, tekivät skriptit aina T-SQL:ää hyödyntäen. Sen ansiosta päivitysskripti pystyy esimerkiksi palauttamaan tehdyt muutokset, jos jossain vaiheessa skriptiä tapahtuu virhe. [10.][11.]

3 Käytetyt työkalut

Insinööriyössä tutkittiin ja käytettiin mm. Microsoftin Visual Studio 2015 -ohjelmankehitystyökalua, Redgaten SQL Compare-, SQL Data Compare- ja ReadyRoll -työkaluja sekä Flexera Softwaren InstallShield-työkalua.

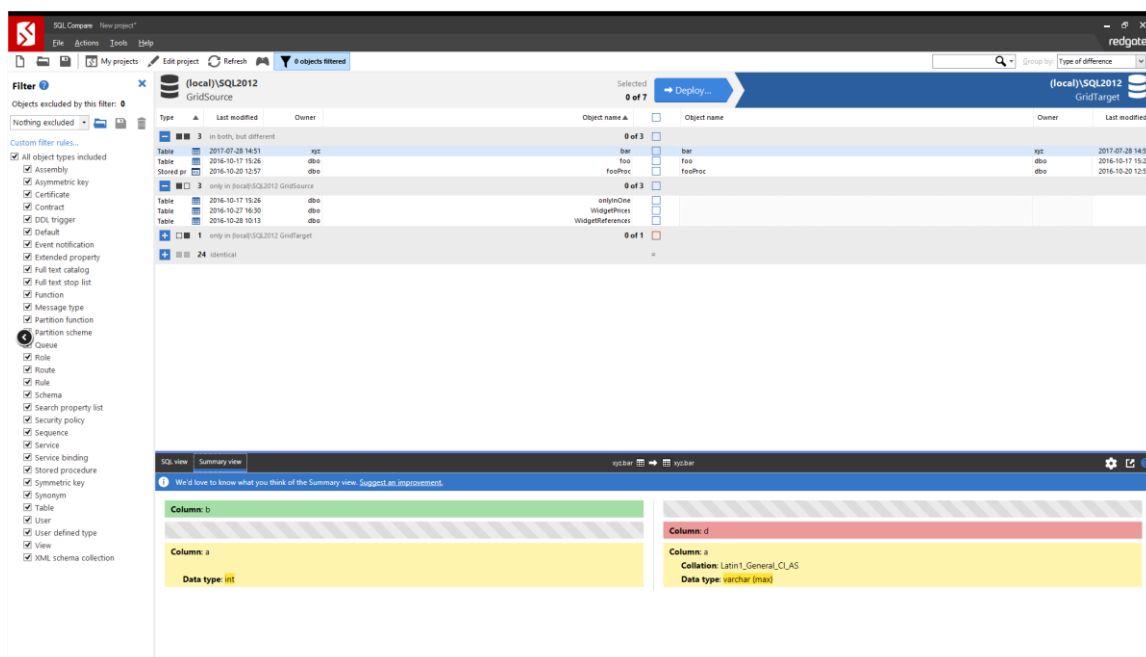
3.1 Microsoft Visual Studio 2015

Microsoftin kehittämä Visual Studio 2015 on ohjelmankehitystyökalu, jossa voi ohjelmoida useilla eri ohjelmointikielillä, mm. Visual Basicilla, C++:lla ja C#:lla [12]. Koska Ponssella käytetään paljon C#-ohjelmointikieltä, tietokantoihin liittyvä olemassa oleva koodi oli kirjoitettu C#:lla ja sitä täytyi insinööriyössä hieman muokata. Visual Studiolla käännettiin tietokantojen päivityksessä hyödynnetty DLL ja siitä oli apua myös, kun DLL ei heti toiminut ja virhettä piti etsiä ajamalla koodia rivi kerrallaan. Sillä pystyy myös tekemään muutoksia tietokantaan sekä tarkastelemaan ja muuttamaan siellä olevaa tietoa, mutta tässä työssä siihen käytettiin enemmän Microsoftin SSMS-työkalua, sillä sen käyttö tuntui helpommalta. Visual Studiota käytettiin myös Redgaten ReadyRoll-työkalun kanssa ja versionhallinnan takia.

3.2 Redgate SQL Compare

Redgate SQL Compare on Redgate Softwaren kehittämä työkalu, ja se soveltuu SQL-tietokantojen ylläpitoon [11]. SQL Comparea ei tosin ole tarkoitettu satojen tietokantojen päivittämiseen, sillä sitä täytyy käyttää käsin. Työkalulla voi kyllä luoda päivitysskriptejä, mutta ReadyRollilla niiden luonti on vaivattomampaa. Redgaten sivuilla työkalusta mainitaankin, että se sopii tilanteeseen, jossa pitää ottaa käyttöön tai tutkia muutoksia kahden SQL-tietokannan välillä [11]. Seuraavalla sivulla kuvassa 2 on SQL Comparen käyttöliittymä, vasemmalla puolella on lähdetietokanta ja oikealla puolella päivitettävä tietokanta, joka päivitetään samanlaiseksi kuin vasemmalla puolella oleva. Käyttäjä voi ennen päivitystä valita käyttöliittymästä tauluja, joita hän ei halua siirrettävän toiseen tietokantaan. Näin työkalulla onnistuu esimerkiksi vain yhden taulun siirtäminen. Tietokantojen alla olevassa osiossa käyttäjä voi tarkistella yksittäisten taulujen välisiä eroja.

Tähän insinöörityöhön SQL Compare ei soveltunut, koska työssä oli vaatimuksena päivityksen automaattisuus ohjelmistopäivityksen yhteydessä. Lisäksi SQL Comparen käyttö vaatisi jonkinlaisen koulutuksen henkilöille, jotka tekevät ohjelmiston päivityksiä, ja heille pitäisi myös hankkia lisenssit työkalun käyttöä varten. Työkalua voi kuitenkin mahdollisesti hyödyntää ongelmatapauksissa, jos esimerkiksi automaattipäivitys ei jostain syystä toimi ja kone täytyy saada päivitettyä. Tällöin nopea ratkaisu päivitykseen voisi olla etäyhteyden ottaminen ongelmakoneeseen ja tietokannan käsin päivittäminen SQL Comparella. Tällaista tilannetta ei tosin saisi tulla ikinä eteen, koska ongelmat harvoin esiintyvät vain yhdessä koneessa, ja pian jonkun työaika kuluisi kokonaan tietokantojen etäpäivityksiin.



Kuva 2. Redgate SQL Compare -työkalu [11].

3.3 Redgate SQL Data Compare

Redgaten SQL Data Compare on yllä mainitun SQL Comparen kaltainen tietokantojen vertailutyökalu, mutta sen sijaan, että verrattaisiin tietokantojen rakennetta, SQL Data Comparella vertaillaan sen nimen mukaisesti tietokannoissa olevia tietoja [13]. Se soveltuu tilanteisiin, joissa pitää synkronoida, siirtää tai vertailla tietoja kahden tietokannan välillä. Insinöörityössä tälle työkalulle ei oikeastaan ollut käyttöä, koska tietokantoja päivittäessä tietojen täytyi pysyä samoina, eikä käsin käytettävä työkalu soveltunut työn vaatimuksiin.

3.4 Redgate ReadyRoll

Myös Redgate ReadyRoll on Redgate Softwaren kehittämä työkalu. Se on suunniteltu SQL-tietokantojen ylläpidon helpottamiseksi. ReadyRoll integroituu suoraan Microsoftin Visual Studioon, ja sillä voi luoda helposti skriptejä, joilla tietokantaa voidaan muokata ilman, että tietokannassa oleva tieto muuttuu. [14.]

Kun tietokannan rakenteeseen tehdään muutos, ReadyRoll vertaa uutta rakennetta aiempaan rakenteeseen ja luo automaattisesti skriptin, jolla vanha rakenne voidaan päivittää uudempaan. Muutoksia voidaan testata oikean tietokannan kopiassa ennen niiden tallennusta, ja näin voidaan huomata ja korjata mahdolliset virheet muutoksissa ennen niiden täytäntöönpanoa. Skriptit voidaan lajitella halutulla tavalla, esimerkiksi versioittain, ja tietokantaa päivitettäessä voidaan valita, mihin versioon tietokanta halutaan päivittää. [14.]

3.5 InstallShield

InstallShield on Flexera Softwaren kehittämä työkalu, jolla voidaan tehdä ohjelmien asennukseen käytettäviä asennuspaketteja [15]. Asennuspakettiin kasataan kaikki ohjelman asennukseen tarvittavat tiedostot ja rekisteriarvot, ja kun ohjelma asennetaan asennuspaketista, asennuspaketti kopioi tiedostot oikeaan paikkaan kohdetietokoneella. Insinööriyössä tietokannan päivitys täytyi saada tapahtumaan InstallShieldillä tehdyn asennuspaketin asennuksen yhteydessä.

InstallShieldissä voi lisätä asennuspakettiin lisätoimintoja ("Custom Action") käyttäen InstallScript-ohjelmointikieltä, tai vaikkapa toisella ohjelmointikielellä käännettyä DLL:ää hyödyntäen. InstallScript tosin eroaa muista ohjelmointikielistä siten, että siinä on paljon valmiita funktioita ja ominaisuuksia, joita voi hyödyntää asennuspaketin tekemisessä.

4 Metsäkoneen mittalaite Ponsse Opti sekä Opti4G

Opti4G on Ponssin kehittämä kuljettajan käyttöliittymä koneen ohjausjärjestelmään. Tässä työssä tietokantojen päivityksen tulee tapahtua Opti4G:n ohjelmistopäivityksen yhteydessä. Kuvassa 3 näkyy metsäkoneen mittalaite, jota nimitetään Optiksi tai Opti-PC:ksi. Opti on myös Ponssin kehittämä, ja siinä on nykyään Windows Embedded Standard 7 -käyttöjärjestelmä, tosin osassa päivitystä vaativissa koneista saattaa olla käytössä vanha Opti, josta löytyy Windows XP. Opti4G:n avulla kuljettaja voi hallita harvesterin asetuksia ja vaikkapa kalibroida koneen. Siinä on myös ajanseuranta-, raportointi-, tiedonsiirto- ja työnhallintatyökalut sekä lisäominaisuutena Ponsse OptiMap2 -karttaohjelma. Edellä mainitut toiminnot hyödyntävät tietokantoja, joiden päivitystä tässä insinööriyössä käsitellään. Kuten kuvassa 3 näkyy, kuljettaja voi säätää esimerkiksi sitä, mikä toiminto missäkin kahvojen napissa on. [16.]



Kuva 3. Ponsse Opti7 kosketusnäytöllä [16]

5 Työn toteutus

Työn toteutus koostui vaatimusten määrittelystä, työvaiheiden suunnittelusta, työkalujen tutkimisesta ja tietokantojen päivityksen demonstroinnista. Työ aloitettiin helmikuussa, ja huhtikuun puolessavälissä oli valmiina malli siitä, miten tietokannat voidaan päivittää.

5.1 Vaatimukset työlle

Vaatimukset työlle olivat, että metsäkoneen ohjausjärjestelmän SQL-tietokannat voidaan päivittää vanhemmista tietokantaversioista samalla, kun metsäkoneen ohjelmaversio päivitetään. Tietokannan päivityksen tuli peruuntua automaattisesti, jos ohjelmaversioon päivitys epäonnistui tai jos kesken tietokannan päivityksen tapahtui virhe. Dataa ei saanut hävitä tietokannasta päivityksen yhteydessä ja tekniikan tuli olla hyvin testattu, ennen kuin sitä voidaan alkaa käyttää tuotantoversioissa.

5.2 Työn suunnittelu

Työn suunnittelu alkoi työkalujen valinnalla. Koska Ponssella oli jo aiemmin kuultu Redgaten työkaluista, päätettiin tutkia, pystyykö insinööriyön toteuttamaan kokonaan niillä, ja harkita vasta myöhemmin muita työkaluja, mikäli niille olisi edelleen tarvetta. Työkaluja päätettiin testata ensin uudella tyhjällä tietokannalla ja sen jälkeen Ponssen kenttätestauksessa olleen metsäkoneen tietokantojen varmuuskopioilla.

5.3 Työn toteutus

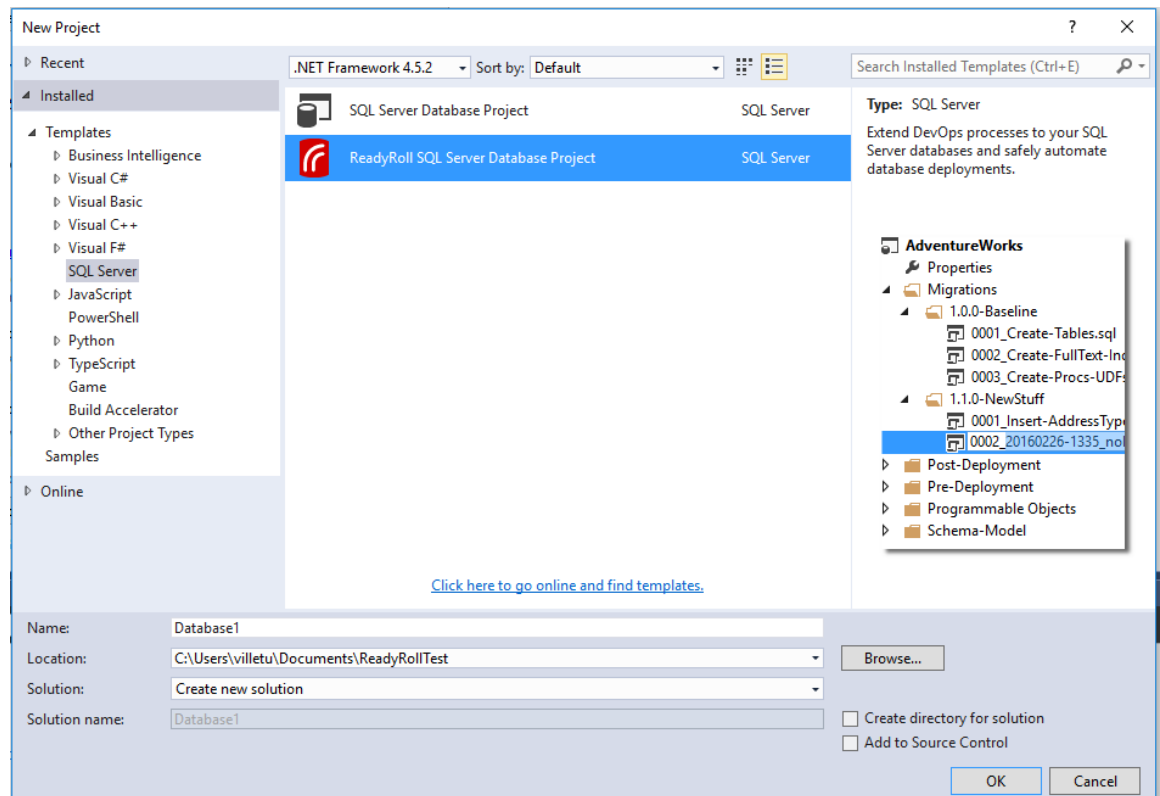
Työ alkoi sillä, että hankittiin kokeiluversio Redgaten työkaluihin. Näin pystyttiin testaamaan ilmaiseksi, mihin työkalut pystyvät. Ensimmäisenä testattiin SQL Comparea ja SQL Data Comparea. Työkaluilla tehtiin erilaisia muutoksia testitietokantaan, jotta nähtiin, mikä kaikki niillä onnistuu. Työkalut osoittautuivat toimiviksi, mutta niissä huomattiin myös varjopuolia. SQL Data Compare osoittautui heti insinööriyön kannalta turhaksi, koska sitä pystyi hyödyntämään oikeastaan vain yhdessä tietokannassa, sen käyttötapaukset olivat aika rajatut ja sitä täytyi käyttää käsin. Vaikka SQL Compare oli

myös käsin käytettävä työkalu, sillä pystyi kuitenkin luomaan päivitysskriptin, jolla pystyi päivittämään vanhasta versiosta uuteen versioon monta tietokantaa. Ongelma tässä työkalussa oli tosin se, että SQL Comparea käytettäessä täytyisi tehdä aina käsin päivitysskripti uusimpaan versioon jokaisen ohjelmistoversion julkaisun yhteydessä. Vaikka tämä olikin yksi vaihtoehto, alettiin etsiä parempaa ja automaattisempaa ratkaisua ongelmaan.

Redgaten työkalujen kokeiluaika meni umpeen kahden viikon kuluttua työn aloittamisesta. Koska SQL Compare oli jo osoittautunut potentiaalisesti ratkaisuksi, päätettiin hankkia Redgatelta yksi oikea lisenssi sen työkaluihin jo tässä vaiheessa. Redgate tarjoaa lisenssejä yksittäisiin työkaluihin tai työkalupaketteihin, joita ovat SQL Toolbelt Essentials, joka sisältää 9 työkalua, ja SQL Toolbelt, joka sisältää 15 työkalua. Koska työkalupakettien välinen hintaero oli vain 900 euroa ja kalliimpi työkalupaketti näytti sisältävän mahdollisesti tähän insinööriyöhön sopivia työkaluja joita SQL Toolbelt Essentialsista ei löytynyt, päätettiin hankkia yksi lisenssi SQL Toolbeltiin. Redgaten lisenssit ovat elinikäisiä, joten 2600 euron kertamaksu koettiin erittäin kohtuulliseksi hinnaksi 15 työkalusta. Normaalisti lisenssin mukana tulee yhden vuoden versiopäivitys- ja käyttötuki, mikä tarkoittaa sitä, että mikäli vuoden kuluessa työkalusta julkaistaan uusi versio, saa senkin käyttöönsä samalla lisenssillä, ja Redgatelta saa tukea työkalujen käyttöön puhelimella, sähköpostilla tai Redgaten asiakasfoorumien kautta. Maksamalla noin 500 euroa enemmän tuki pitenee kahteen vuoteen tai maksamalla noin 1000 euroa enemmän tuki pitenee kolmeen vuoteen. Tässä tapauksessa tyydyttiin yhden vuoden mittaiseen tukeen.

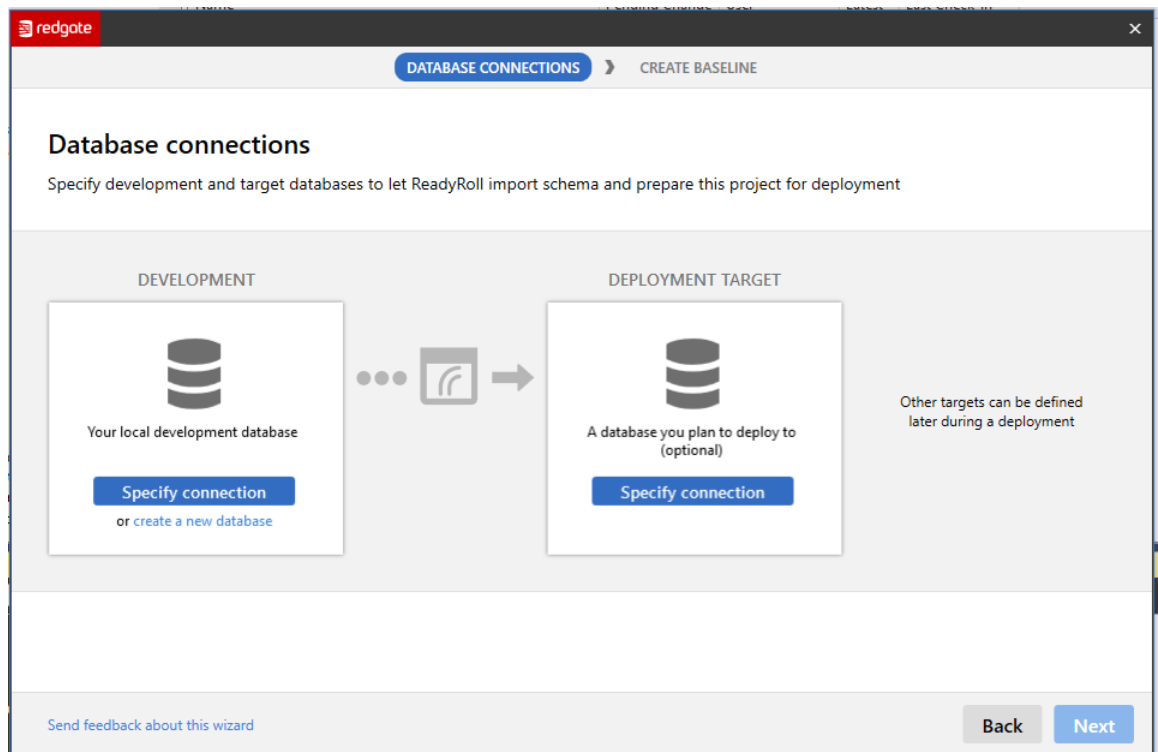
Redgatelta päätettiin myös kysyä, mikä olisi sen mielestä paras työkalu tähän työhön. Redgatelta vastattiin, että ReadyRollilla pitäisi pystyä tekemään juuri työltä vaaditut asiat. Tilatun lisenssin saavuttua alettiin tutkia, mitä ReadyRollilla voidaan tehdä. Heti huomattiin, että ReadyRollin käyttöä varten Visual Studiossa täytyy luoda uusi ReadyRoll-projekti, eikä tavallista Visual Studiolla luotua Database-projektia voi automaattisesti muuttaa ReadyRoll-projektiksi. Seuraavaksi täytyi tutkia ReadyRoll-projektin luontia Visual Studiossa.

ReadyRoll-projektin pystyy luomaan valitsemalla Visual Studion yläreunasta File → New → Project. Sen jälkeen näytölle aukeaa alla näkyvän kuvan 4 mukainen projektin luonti-ikkuna. Navigoidaan vasemmasta palkista Installed → Templates → SQL Server ja sieltä valitaan normaalin SQL Server Database -projektin alle ilmestynyt ReadyRoll SQL Server Database -projekti.



Kuva 4. ReadyRoll SQL Server Database projektin luonti-ikkuna

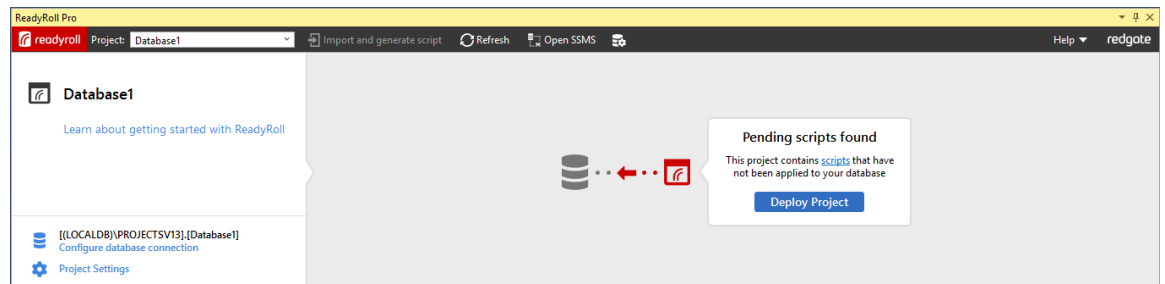
Kun ReadyRoll-projekti luodaan, avautuu projektin määrittelyikkuna. Sen jälkeen esiin tulee olevan kuvan 5 mukainen ikkuna, jossa käyttäjä voi valita "development databaseksi" eli kehitystietokannaksi olemassa olevan tietokannan tai luoda uuden tietokannan, johon tietokantamuutokset jatkossa tehdään, ja "deployment target"-tietokannan, jota käytetään nyt projektia luodessa ensimmäisten pohjatietokantaskriptien luomiseen. Deployment target -tietokantaa ei tarvitse valita, jos halutaan luoda täysin uusi ja tyhjä tietokanta. Seuraavaksi painetaan "Next", ja sen jälkeen "Create Baseline". Sen jälkeen ReadyRoll luo pohjan valitun "deployment target"-tietokannan perusteella, ja tietokantamuutosten tekemisen voi aloittaa valittuun kehitystietokantaan.



Kuva 5. ReadyRoll-projektin alustusikkuna

Tässä työssä oli valmis tietokanta, jota haluttiin päivittää, ja siksi ReadyRoll-projektin luontivaiheessa "deployment targetiksi" valittiin olemassa oleva tietokanta. ReadyRoll-projektin luonti kestää muutaman minuutin deployment targetina olleen tietokannan rakenteesta riippuen. Näin luotuun ReadyRoll-projektiin saatiin siirrettyä tietokanta vanhasta SQL Server Database -projektista. Kehitystietokannaksi luotiin tietokoneelle paikallinen tietokanta, jossa muutoksia voitaisiin rauhassa testata.

ReadyRoll-projektin luomisen jälkeen testattiin päivitysskriptien luontia. Se onnistui ilman vaikeuksia. Kun projekti luodaan, Visual Studion alareunaan ilmestyy alla olevan kuvan 6 mukainen ikkuna. Deploy Project -nappia painettaessa ReadyRoll luo .sql-muotoisen päivitysskriptin, jolla voidaan luoda tietokanta, joka yllä valittiin kehitystietokannaksi.



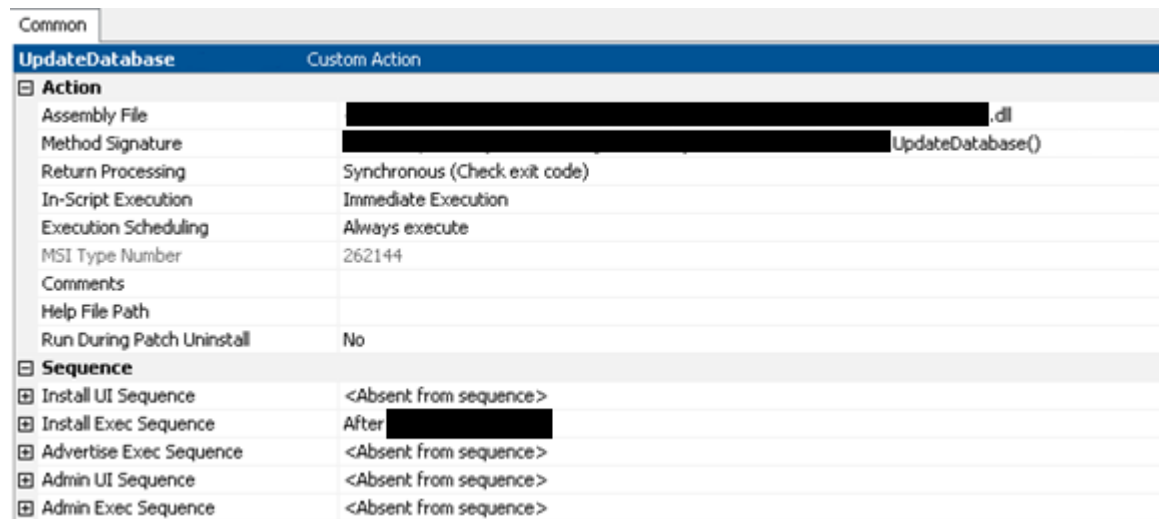
Kuva 6. Visual Studion ReadyRoll-ikkuna

Seuraavaksi testattiin päivitysskriptin suorittamista ohjelmiston asennuksen yhteydessä. Sitä varten tehtiin InstallShieldillä testiasennuspaketti, jotta välttyttiin sotkemasta tuotannossa käytettäviä paketteja. Testiasennuspaketti kopioitiin suoraan tuotantoasennuspaketista, ja siihen lisättiin yksi Custom Action, jossa tietokannan päivitysskripti suoritetaan. Tässä käytettiin apuna ennestään löytynyttä C#:lla tehtyä koodia.

Koodista löytyi valmiiksi keino suorittaa .sql-tiedostoja, joten muutokset olivat yksinkertaisia. Täytyi vain lisätä koodiin uusi funktio UpdateDatabase(), jossa ReadyRollin generoima .sql-tiedosto suoritetaan. Esimerkki funktiosta löytyy liitteestä 1. Esimerkkifunktion sqlQuery-muuttuja on ReadyRollilla generoitu päivitysskripti string-muuttujana. Työssä päivitysskripti lisättiin Visual Studion projektiin resurssina ja näin se saatiin helposti muunnettua string-muotoon. Toinen muuttuja, "connectionString", on merkkijono, jolla voidaan kirjautua sisään tietokantaan. Lisätietoa niistä löytyy lähteestä 17.

InstallShieldin Custom Action laitettiin kutsumaan UpdateDatabase-funktiota DLL:n kautta. Tehty Custom Action oli alla olevan kuvan 7 mukainen. Custom Actionin suoritusehdoksi laitettiin "REMOVE <> "ALL"", joka tarkoittaa, että Custom Action suoritetaan aina, paitsi sovelluksen poiston yhteydessä. InstallShieldissä ohjelman voi asentaa, ohjelmalle voi suorittaa pienen tai suuren päivityksen tai ohjelman asennuksen voi korjata, ja edellä mainitulla ehdolla Custom Action suoritetaan kaikissa edellä mainituissa tapauksissa. Custom Actionin suorittaminen monta kertaa samalle tietokannalle ei ole haitallista, koska ReadyRollin generoima päivitysskripti luo tietokantaan taulun, johon tallennetaan, mitä päivityksiä tietokannassa on ja missä

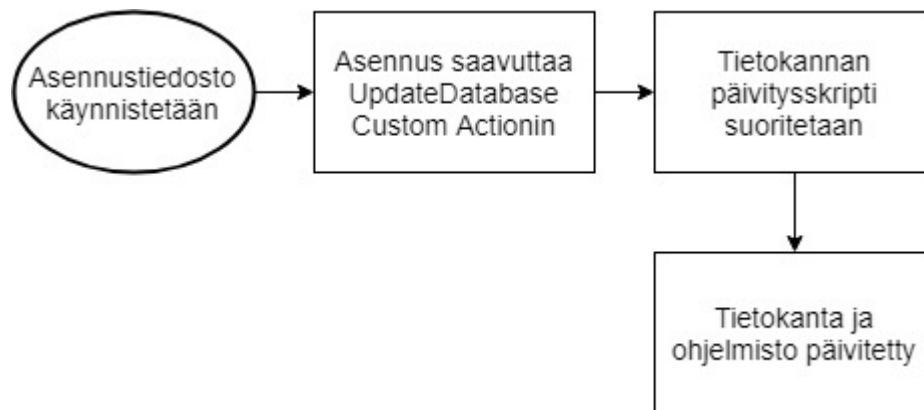
versiossa tietokanta on, ja mikäli tietokanta on jo uusimmassa mahdollisessa versiossa, eikä siihen tarvitse tehdä muutoksia, päivitysskripti katsoo, että tietokanta on kunnossa, eikä muuta tapahdu. Aikaa kaikkeen tuohon kuluu vain muutama sekunti, eikä siitä ole haittaa, kun asennus kestää muutamia minuutteja.



Kuva 7. UpdateDatabase Custom Action InstallShieldissä

Yhtenä työn vaatimuksena oli, että jos ohjelmistopäivitys epäonnistuu, tietokantapäivityksen tulee myös peruuntua ja toisinpäin. Näitä ominaisuuksia ei saatu vielä toteutettua. Työssä tehdyssä esimerkissä tietokanta jää päivitettyyn versioon, jos asennuspaketissa tapahtuu virhe UpdateDatabase Custom Actionin jälkeen. Toisaalta UpdateDatabase suoritetaan asennuksen lopussa, joten todennäköisyys, että virhe tapahtuu sen suorituksen jälkeen, on pieni. Jos asennuspaketissa tapahtuu virhe ennen kuin päästään tietokannan päivittävään Custom Actioniin, tietokannan päivitystä ei suoriteta. Jos Custom Actionin aikana eli käytännössä UpdateDatabase-funktiossa tai tietokannan päivitysskriptissä tapahtuu virhe, tietokannan päivitys keskeytyy, mutta asennuspaketti jatkaa asennusta ja ohjelmistopäivitys suoritetaan. Työ vaatii vielä jatkokehitystä, jotta kaikki vaatimukset saadaan täytettyä.

Yhteenveto päivitysprosessista ideaalitalanteessa näkyy kuvassa 8. InstallShieldillä tehty asennustiedosto käynnistetään metsäkoneen mittalaitteella ja se alkaa asentaa ohjelmistopäivitystä. Kun asennus tulee Custom Actionin kohdalle, jossa kutsutaan UpdateDatabase-funktiota, tietokanta päivitetään. Lopputuloksena mittalaitteella on päivitetty ohjelmaversio ja päivitetty tietokantaversio. Mikäli missään vaiheessa asennusprosessia tapahtuu virhe, prosessi keskeytyy ja kaikki tehdyt muutokset kumotaan.



Kuva 8. Asennusprosessi ideaalitalanteessa.

5.4 Työn tulokset

Opinnäytetyön tavoitteena oli löytää keino päivittää metsäkoneessa olevia SQL-tietokantoja automaattisesti metsäkoneen ohjausjärjestelmän ohjelmistopäivityksen yhteydessä. Tuloksena saatiin aikaan esimerkki siitä, miten päivitys onnistuu, mutta työ vaatii vielä hieman jatkokehitystä, jotta kaikki ongelmakohtat saadaan ratkottua ja tekniikka otettua käyttöön Ponssen tuotekehityksessä. Testaamisen aikana tietokannasta paljastui ongelma, joka estää automaattipäivityksen käyttöönoton, kunnes se korjataan. Ongelma oli se, että eräiden tietokantaobjektien nimet olivat erilaiset jokaisessa kannassa, joten niitä on mahdotonta käsitellä, koska se tapahtuu skriptissä aina nimen perusteella. Työn kesken jäämiseen vaikutti osittain myös se, että työssä pitää vielä kuunnella lisää mielipiteitä tietokantakehittäjiltä, mikä menetelmä heidän mielestään on paras, sillä työn tulos vaikuttaa heidän työhönsä eniten. Kokousten järjestäminen vie aina oman aikansa, kun kaikkien osallistujien aikataulut täytyy ottaa huomioon.

Kun tyhjään testitietokantaan tehtiin muutoksia ReadyRollilla luodulla skriptillä, ne onnistuivat aina. Haasteita alkoi esiintyä, kun testitietokantaan lisättiin tietoa. Tällöin hankaluuksia tuli erityisesti rajoitteisiin tehtyjen muutosten kanssa. Ongelmallisia muutoksia olivat esimerkiksi kentän maksimipituuden lyhentäminen, joka ei onnistunut, jos tietokannassa oli uutta maksimipituutta pidempää tietoa. Täytyy tutkia vielä lisää, olisiko olemassa jokin asetus, jolla ReadyRollin skripti lyhentäisi tietokannassa olevan tiedon uuden maksimipituuden mittaiseksi automaattisesti. Toinen esimerkki ongelmallisesta muutoksesta oli tietotyyppin muutos. Jos string-kenttää yritti muuttaa int-kentäksi, mutta tietokannassa kaikki siihen kenttään tallennettu tieto ei ollut numeroina, muutos ei onnistunut, koska ReadyRoll ei halunnut tuhota tekstimuotoista tietoa eikä sitä voinut muuttaa numeromuotoon.

Jatkokehityksenä työlle tulisi tutkia, mikä on paras keino suorittaa ongelmalliset tietokantamuutokset. Yksi vaihtoehto on ainakin tiedon muokkaaminen ReadyRollilla luodun skriptin osiossa, joka suoritetaan ennen kuin muutoksia aletaan tekemään. Näin muutosten kohteena olevien kenttien tieto olisi aina uusien rajoitteiden mukaista. Tämä tosin vaatii jonkin verran käsityötä, kun skriptiä luodaan. Toisaalta voi olla mahdollista, ettei tällaisia rajoitteisiin kohdistuvia muutoksia tai kenttien tietotyyppien muutoksia tehdä kovin usein tai välttämättä ikinä, jolloin käsityötä skriptien eteen ei tarvitse tehdä usein. Jatkokehityksessä täytyy myös pohtia, miten ReadyRoll otetaan Ponssen tuotekehityksessä käyttöön; korvataanko nykyinen SQL Server Database -projekti sillä vai aletaanko sen rinnalla ylläpitää ReadyRoll-projektia, joka päivitetään aina, kun

ohjelmistoversioon tulee päivitys. Mikäli nykyinen projekti korvataan ReadyRoll-projektilla, kaikilla tietokantakehittäjillä tulee olla lisenssi ReadyRollin käyttöön. ReadyRoll-lisenssiä ei voi ainakaan Redgaten sivuilta ostaa erikseen, mutta sen saa, kun ostaa Redgaten SQL Toolbeltin tai Visual Studio 2017 Enterprise -version.

Työn vaatimuksia ajatellen on hyvä, ettei ReadyRoll oletuksena tuhoa tietokannan tietoja muutoksia tehdessään. Asiakkaiden tietoja ei tule tuhota oletuksena, sillä aina on pieni mahdollisuus, että tuotantoon pääsee lipsahtamaan muutos, jonka vaikutuksia ei ole huomioitu. Tietokannan kehittäjille se tuo lisätyötä, sillä kaikki muutokset täytyy testata perin pohjin, ja vielä tarvittaessa kirjoittaa skriptiin logiikka, joka käsittelee tiedon muuttamisen sopivaksi, tai muuten alkaa esiintyä tilanteita, joissa päivitys ei toimi kaikkiin koneisiin. Kenties paras tapa testata tietokantaan tehtyjä muutoksia on testata niitä kenttätestauksessa tai asiakkaalla olleen metsäkoneen tietokannan varmuuskopiolla, koska tällöin tietokanta vastaa oikeaa päivitystilanteessa olevaa tietokantaa. Tämän lisäksi olisi kuitenkin hyvä lisätä tietokantaan tahallaan vääränmallista tietoa, jotta nähdään, miten päivitys silloin käyttäytyy. Jos päivitys estyy, täytyy miettiä, tulisiko varmuuden vuoksi lisätä skriptiin vääränlaisen tiedon poistava tai muokkaava osio, vaikka tällaista tietoa ei koskaan kentällä tietokannassa esiintyisikään. Näin saataisiin varmasti mahdollisimman paljon virheitä ennaltaehkäistyä.

Työn aikana ei saatu täytettyä sitä vaatimusta, että jos tietokannan päivitysskriptissä tapahtuu virhe, myös ohjelmistopäivityksen tulee peruuntua. Tehdyssä demonstraatiossa tietokannan päivitysskriptin virhetilanteessa tietokannan päivitys peruuntui, mutta ohjelmistopäivitys suoritettiin loppuun. Jatkokehityksenä täytyy tehdä UpdateDatabase-funktioon ominaisuus, joka huomaa, jos tietokannan päivityksessä tapahtuu virhe, ja ilmoittaa sen eteenpäin asennusprosessille, jotta se osaa keskeyttää myös ohjelmiston päivityksen.

Työn aikana tutkittiin ja hankittiin Redgaten SQL Toolbelt -lisenssi ja myös muista sen mukana tulleista työkaluista ja niiden testaamisesta ja tutkimisesta voi olla Ponsselle hyötyä tulevaisuudessa. SQL Compare ja SQL Data Compare voivat auttaa Ponssen palvelimilla sijaitsevien tietokantojen ylläpidossa ja muutoksissa.

Lähteet

- (1) Ponsse Oyj. Saatavilla: <http://www.ponsse.com/fi/ponsse> Luettu 15.4.2018
- (2) SQL | computer language. Saatavilla: <https://www.britannica.com/technology/SQL> Luettu 25.3.2018
- (3) Ollikainen V. Osa 1: Tietokanta. Saatavilla: https://wiki.jyu.fi/download/attachments/14849169/1_tietokanta.ppt?version=1&modificationDate=1320223719000&api=v2 Luettu 22.3.2018
- (4) U.S. Department of Transportation. Data Integration Glossary. Saatavilla: <https://www.fhwa.dot.gov/infrastructure/asstmgmt/010394.pdf> Luettu 16.4.2018
- (5) SQL Constraints. Saatavilla: https://www.w3schools.com/sql/sql_constraints.asp Luettu 16.4.2018
- (6) SQL Stored Procedures. Saatavilla: <https://docs.microsoft.com/en-us/sql/relational-databases/stored-procedures/stored-procedures-database-engine?view=sql-server-2017> Luettu 16.4.2018
- (7) SQL - Indexes. Saatavilla: <https://www.tutorialspoint.com/sql/sql-indexes.htm> Luettu 16.4.2018
- (8) Microsoft SQL Server Editions. Saatavilla: <https://www.microsoft.com/fi-fi/sql-server/sql-server-2017-editions> Luettu 16.4.2018
- (9) SQL Server Management Studio. Saatavilla: <https://docs.microsoft.com/en-us/sql/ssms/download-sql-server-management-studio-ssms?view=sql-server-2017> Luettu 16.4.2018
- (10) Rouse M, Jones S, Hughes A. T-SQL (Transact-SQL). Saatavilla: <https://searchsqlserver.techtarget.com/definition/T-SQL> Luettu 16.4.2018
- (11) Redgate SQL Compare. Saatavilla: <https://www.red-gate.com/products/sql-development/sql-compare/index> Luettu 13.4.2018
- (12) Microsoft Visual Studio. Saatavilla: <https://www.visualstudio.com/vs/> Luettu 16.4.2018
- (13) Redgate SQL Data Compare. Saatavilla: <https://www.red-gate.com/products/sql-development/sql-data-compare/> Luettu 13.4.2018
- (14) Redgate ReadyRoll. Saatavilla: <https://www.red-gate.com/products/sql-development/readyroll> Luettu 16.4.2018
- (15) InstallShield. Saatavilla: <https://www.flexera.com/producer/products/software-installation/installshield-software-installer/> Luettu 16.4.2018
- (16) Ponsse Opti. Saatavilla: http://www.ponsse.com/fi/content/download/3879/111673/file/PONSSE_Opti_FIN.pdf Luettu 2.4.2018
- (17) SQL Server connection strings. Saatavilla: <https://www.connectionstrings.com/sql-server/> Luettu 17.4.2018

Liitteet

Liite 1

Esimerkki UpdateDatabase-funktiosta:

```
void UpdateDatabase()
{
    ExecuteSqlScript(sqlQuery, connectionString);
}

private static void ExecuteSqlScript(string sqlQuery, string
connectionString)
{
    using (SqlConnection connection = new SqlConnection(
        connectionString))
    {
        SqlCommand command = new SqlCommand(sqlQuery, connection);
        command.Connection.Open();
        command.ExecuteNonQuery();
    }
}
```